



# Query-driven Repairing of Inconsistent DL-Lite Knowledge Bases

Meghyn Bienvenu, Camille Bourgaux, François Goasdoué

## ► To cite this version:

Meghyn Bienvenu, Camille Bourgaux, François Goasdoué. Query-driven Repairing of Inconsistent DL-Lite Knowledge Bases. DL: Description Logics, Apr 2016, Le Cap, South Africa. hal-02152060

**HAL Id: hal-02152060**

**<https://hal.archives-ouvertes.fr/hal-02152060>**

Submitted on 11 Jun 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Query-driven Repairing of Inconsistent DL-Lite Knowledge Bases (Extended Abstract)

Meghyn Bienvenu<sup>1</sup>, Camille Bourgaux<sup>2</sup>, and François Goasdoué<sup>3</sup>

<sup>1</sup> CNRS, Univ. Montpellier, Inria, Montpellier, France

<sup>2</sup> Univ. Paris-Sud, CNRS, Orsay, France <sup>3</sup> Univ. Rennes 1, CNRS, Lannion, France

## 1 Query-driven repairing

Ontology-mediated query answering (OMQA) is a recent approach to data access in which conceptual knowledge provided by an ontology is exploited when querying incomplete data (see [6] for a survey). As efficiency is a primary concern, significant research efforts have been devoted to identifying ontology languages with favorable computational properties. The DL-Lite family of description logics [8], which underlies the OWL 2 QL profile [20], has garnered significant interest as it allows OMQA to be reduced to standard database query evaluation.

Beyond efficiency, it is important for OMQA systems to be robust to inconsistencies stemming from errors in the data. Inspired by work on consistent query answering in databases [3], several inconsistency-tolerant semantics have been developed for OMQA, with the aim of providing meaningful answers in the presence of inconsistencies. Of particular relevance to the present paper are the *brave semantics* [7], which returns all query answers that have some *cause* (i.e. minimal consistent subset that supports the answer), and the more conservative *IAR semantics* [17] that requires that facts in the cause not belong to any *conflict* (i.e. minimal inconsistent subset). Both semantics have appealing computational properties: for most DL-Lite dialects, including DL-Lite<sub>R</sub>, conjunctive query answering is tractable in data complexity [18, 7].

While inconsistency-tolerant semantics are essential for returning useful results when consistency cannot be achieved, they by no means replace the need for tools for improving data quality. This extended abstract presents our work [5] that proposes a complementary approach that exploits user feedback about query results to identify and correct errors. We consider the following scenario: a user poses conjunctive queries against a possibly inconsistent DL-Lite<sub>R</sub> knowledge base (KB) and receives the sets of possible answers (i.e. those holding under brave semantics) and almost sure answers (those holding under IAR semantics). When examining the results, he detects some *unwanted answers*, which should not have been retrieved, and identifies *wanted answers*, which should be present. Ideally, the unwanted tuples should not be returned as possible answers, and all of the desired tuples should be found among the sure answers.

A *query-driven repairing problem (QRP)* consists of a KB to repair and two sets of Boolean conjunctive queries that the KB should entail or not entail. A *repair plan* is a pair of sets of assertions to delete or add to the ABox. It addresses

all defects if the resulting KB is such that every wanted answer holds under IAR semantics and every unwanted answer does not hold under brave semantics.

There are several reasons to use queries to guide the repairing process. First, we note that it is typically impossible (for lack of time or information) to clean the entire dataset, and therefore reasonable to focus the effort on the parts of the data most relevant to users’ needs. In the database arena, this observation inspired work on integrating entity resolution into the querying process [1]. Second, expert users may have a good idea of which answers are expected for queries concerning their area of expertise, and thus queries provide a natural way of identifying flaws. Indeed, [16] recently proposed to use queries to search for errors and help evaluate linked data quality. Finally, even non-expert users may notice anomalies when examining query results, and it would be a shame to not capitalize on this information, and in this way, help distribute the costly and time-consuming task of improving data quality as argued in [2].

The problem of modifying DL KBs to ensure (non)entailments of assertions and/or axioms has been investigated in many works, see e.g. [11, 9, 13]. Our framework is inspired by that of [15, 14], in which a user specifies two sets of axioms that should be entailed or not by a KB and repair plans are pairs of sets of axioms to remove and add to obtain an ontology satisfying these requirements.

## 2 Optimal repair plans

If we want to avoid introducing new errors, a fully automated repairing process is impossible: we need the user to validate every assertion that is removed or added in order to remove (resp. add) only assertions that are false (resp. true). We formalize the user’s knowledge by a set of models of the TBox and say that an assertion is true (resp. false) if it is true (resp. false) in every of these models, unknown otherwise. A repair plan is *validatable* if it removes (resp. add) only assertions that are known to be false (resp. true).

As validatable repair plans addressing all defects are not guaranteed to exist (for instance, if the user knows that some answer is wrong but cannot pinpoint which assertion is at fault), our aim will be to find validatable repair plans that are optimal in the sense that they address as many of the defects as possible.

We compare repair plans based on the answers they *satisfy*, an unwanted answer being satisfied if it does not hold under brave semantics in the resulting KB, and a wanted answer being satisfied if it has a cause without any conflict and that does not contain any assertion known to be false (i.e. if it holds under IAR semantics ‘for a good reason’). We obtain five preference relations, taking into account one or both criteria, combined according to the Pareto principle or the lexicographic method, and consider global or local optimality for each.

In order to gain a better understanding of the computational properties of the different ways of ranking repair plans, we study the complexity of deciding if a given repair plan is optimal w.r.t. the different criteria. Our complexity analysis reveals that the notions of global optimality based upon the preference relations that take into account both criteria have undesirable computational properties:

even when provided with all relevant user knowledge, it is intractable to decide whether a given plan is optimal. Moreover, while plans globally optimal regarding unwanted (resp. wanted) answers can be interactively constructed in a monotonic fashion by removing further false assertions (resp. and adding further true assertions), building a globally optimal plan for a preference relation that involves both unwanted and wanted answers may require backtracking over answers already satisfied. For the preceding reasons, we target validatable repair plans that are both *globally optimal* for unwanted or wanted answers (depending which is preferred) and *locally optimal* for the Pareto preference. We give interactive algorithms for building such repair plans by cleaning the relevant part of the data, then inserting assertions to create causes for wanted answers that are not satisfied while preserving already satisfied answers.

For future work, when insertions are needed, it would be helpful to provide users with suggestions of assertions to add. The framework of query abduction [10], recently extended to inconsistent KBs [12], could provide a starting point.

### 3 Optimal deletion-only repair plans

A repair plan is *deletion-only* when it contains no insertion. In this simpler setting, the previously introduced notions of optimality collapse. It is possible to further assist the user by taking advantage of the fact that subsets of the ABox to remove to address all defects can be automatically identified, then interactively transformed into optimal repair plans. We call such subsets potential solutions.

We call an assertion *relevant* if it appears in a cause of some answer or in the conflicts of a cause of some wanted answer. If an assertion  $\alpha$  appears in every potential solution, either it is false, or there is no validatable potential solution. We call such assertions *necessarily false*. If  $\alpha$  appears in no potential solution, it is necessary to keep it to retrieve some wanted answers under IAR semantics, so either it is not false, or it is not possible to satisfy all wanted answers. We call such assertions *necessarily nonfalse*. We believe that validating sets of necessarily (non)false assertions requires less effort than hunting for false assertions among relevant ones. Thus we propose an algorithm for constructing an optimal deletion-only plan that exploits this idea and uses SAT encodings to compute the assertions to present to the user. Borrowing ideas from work on reducing user effort in interactive revision, e.g. [19, 21], we use a notion of *impact* to determine the order of presentation of relevant assertions. We also study the complexity of the related decision problems.

We made preliminary experiments on core components of the algorithm. We use the CQAPri benchmark [4] (available at [www.lri.fr/~bourgaux/CQAPri](http://www.lri.fr/~bourgaux/CQAPri)) to build 26 QRPs, with 8 to 121 answers. In our experiments, deciding if a potential solution exists, and computing the relevant assertions, takes a few milliseconds. The difficulty of computing the necessarily (non)false assertions correlates with the number of relevant assertions induced by QRPs and the observed times seem reasonable in practice. Ranking the remaining relevant assertions based on their impact is time consuming, so we plan to optimize impact computation.

*Acknowledgements* This work was supported by contract ANR-12-JS02-007-01.

## References

1. Altwaijry, H., Kalashnikov, D.V., Mehrotra, S.: Query-driven approach to entity resolution. *Proceedings of the VLDB Endowment (PVLDB)* 6(14), 1846–1857 (2013)
2. Bergman, M., Milo, T., Novgorodov, S., Tan, W.: QOCO: A query oriented data cleaning system with oracles. *PVLDB* 8(12), 1900–1911 (2015), <http://www.vldb.org/pvldb/vol8/p1900-bergman.pdf>
3. Bertossi, L.E.: *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2011)
4. Bienvenu, M., Bourgaux, C., Goasdoué, F.: Explaining inconsistency-tolerant query answering over description logic knowledge bases. In: *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)* (2016)
5. Bienvenu, M., Bourgaux, C., Goasdoué, F.: Query-driven repairing of inconsistent DL-Lite knowledge bases. In: *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)* (2016)
6. Bienvenu, M., Ortiz, M.: Ontology-mediated query answering with data-tractable description logics. In: *Lecture Notes of the 11th International Reasoning Web Summer School. LNCS*, vol. 9203, pp. 218–307. Springer (2015)
7. Bienvenu, M., Rosati, R.: Tractable approximations of consistent query answering for robust ontology-based data access. In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)* (2013)
8. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning (JAR)* 39(3), 385–429 (2007)
9. Calvanese, D., Kharlamov, E., Nutt, W., Zheleznyakov, D.: Evolution of DL-Lite knowledge bases. In: *Proceedings of the 9th International Semantic Web Conference (ISWC)* (2010)
10. Calvanese, D., Ortiz, M., Simkus, M., Stefanoni, G.: Reasoning about explanations for negative query answers in DL-Lite. *Journal of Artificial Intelligence Research (JAIR)* 48, 635–669 (2013)
11. De Giacomo, G., Lenzerini, M., Poggi, A., Rosati, R.: On instance-level update and erasure in description logic ontologies. *Journal of Logic and Computation* 19(5), 745–770 (2009)
12. Du, J., Wang, K., Shen, Y.: Towards tractable and practical ABox abduction over inconsistent description logic ontologies. In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)* (2015)
13. Gutierrez, C., Hurtado, C.A., Vaisman, A.A.: RDFS update: From theory to practice. In: *Proceedings of the 8th European Semantic Web Conference (ESWC)* (2011)
14. Jiménez-Ruiz, E., Grau, B.C., Horrocks, I., Llavori, R.B.: Ontology integration using mappings: Towards getting the right logical consequences. In: *Proceedings of the 6th European Semantic Web Conference (ESWC)* (2009)
15. Jiménez-Ruiz, E., Grau, B.C., Horrocks, I., Llavori, R.B.: Supporting concurrent ontology development: Framework, algorithms and tool. *Data & Knowledge Engineering Journal (DKE)* 70(1), 146–164 (2011)
16. Kontokostas, D., Westphal, P., Auer, S., Hellmann, S., Lehmann, J., Cornelissen, R., Zaveri, A.: Test-driven evaluation of linked data quality. In: *Proceedings of the 23rd International Conference on World Wide Web (WWW)* (2014)

17. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant semantics for description logics. In: Proceedings of the 4th International Conference on Web Reasoning and Rule Systems (RR) (2010)
18. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Query rewriting for inconsistent DL-Lite ontologies. In: Proceedings of the 5th International Conference on Web Reasoning and Rule Systems (RR) (2011)
19. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Supporting manual mapping revision using logical reasoning. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI) (2008)
20. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language profiles. W3C Recommendation (11 December 2012), available at <http://www.w3.org/TR/owl2-profiles/>
21. Nikitina, N., Rudolph, S., Glimm, B.: Interactive ontology revision. *Journal of Web Semantics (JWS)* 12, 118–130 (2012)